Introduction to Data Science – Week 5

HEZI RESHEFF

COURSE 55793

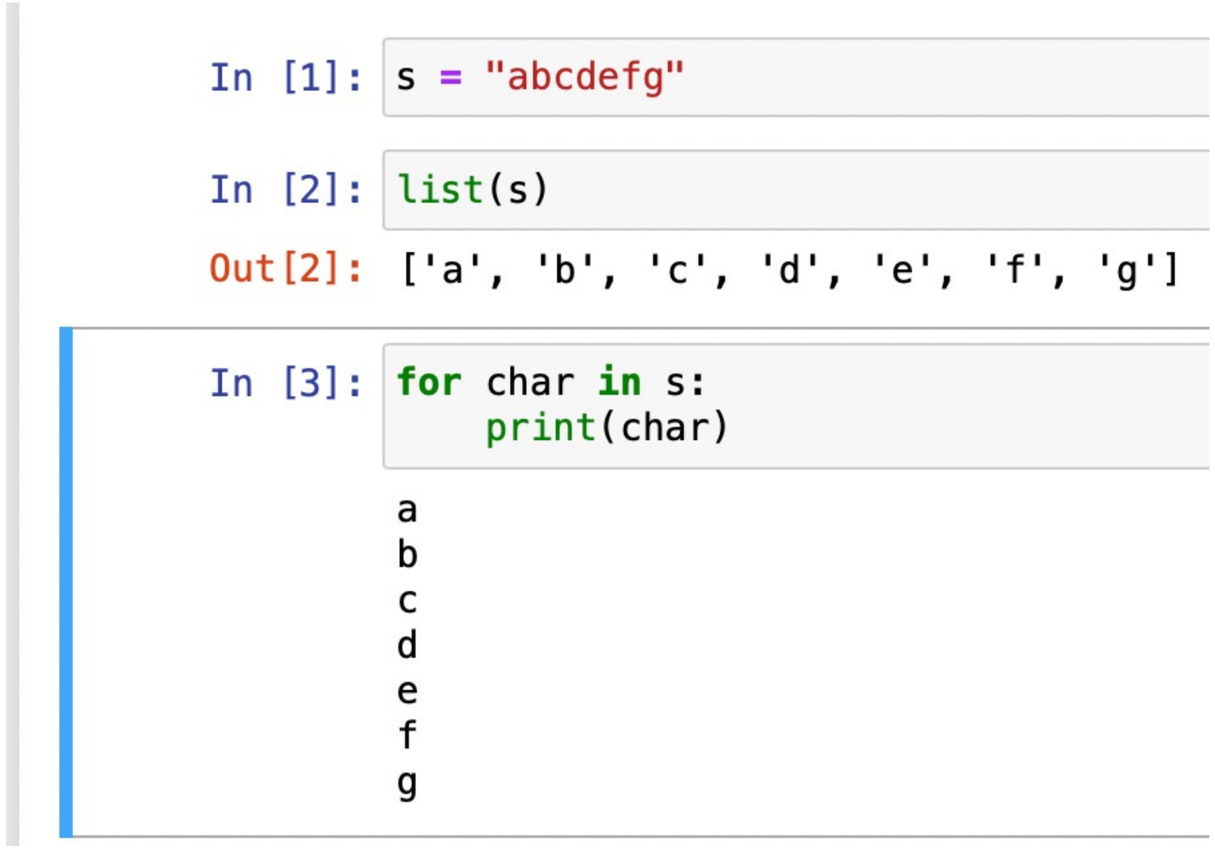
2022-2023

Lessons 1-4 recap

- data types: int, float, bool, list, set, tuple, dict
- if statements
- iterators and iterability
- loops: for, while
- functions

A note about the exercise

- strings are iterable.

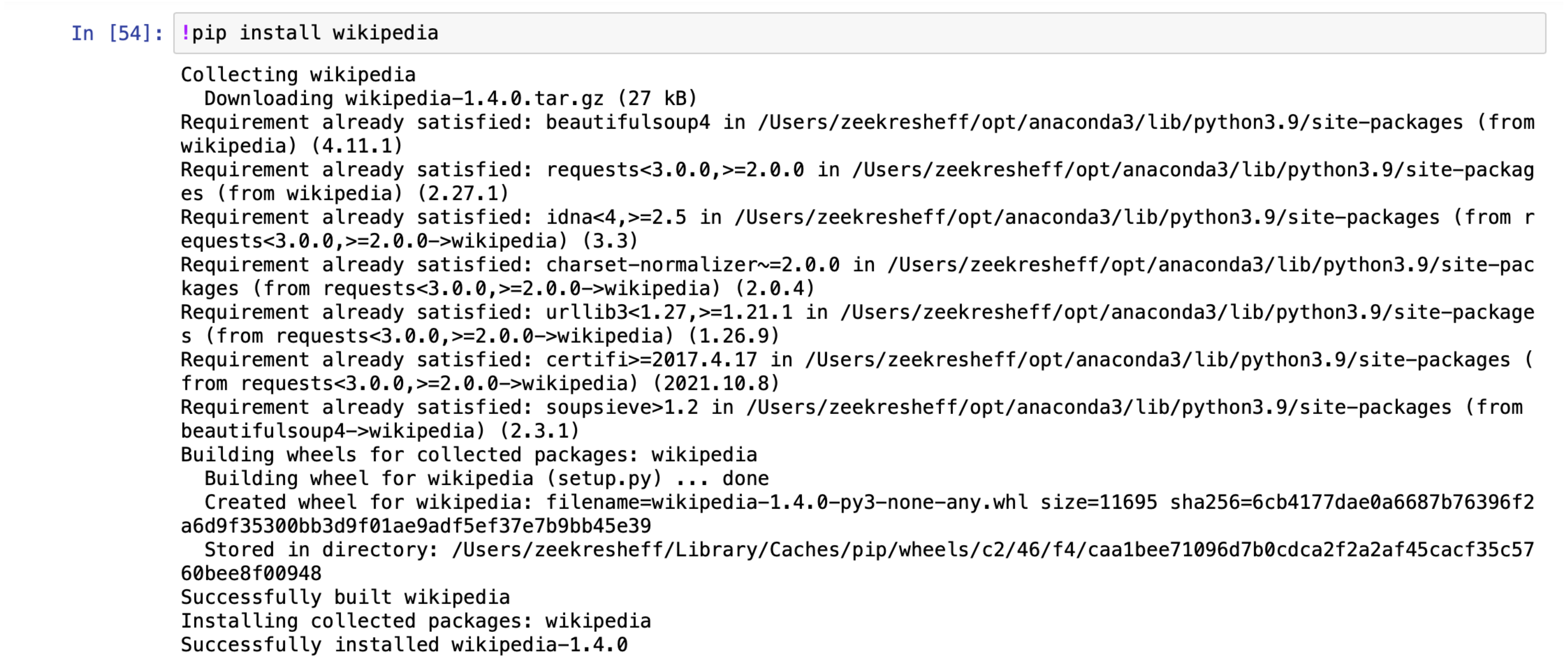


```
In [1]: s = "abcdefg"

In [2]: list(s)

Out[2]: ['a', 'b', 'c', 'd', 'e', 'f', 'g']

In [3]: for char in s:
            print(char)

        a
        b
        c
        d
        e
        f
        g
```

This week

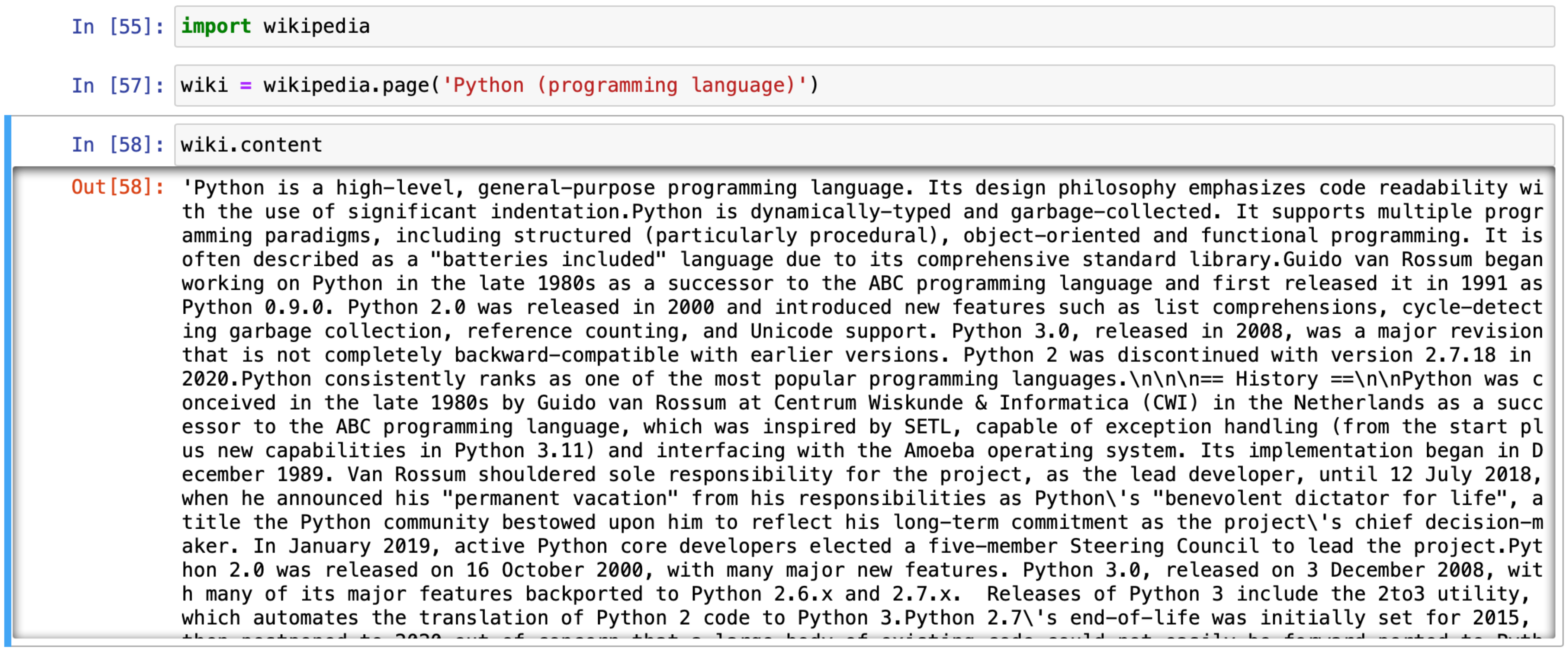
- Finishing up dict (sorting by key or value…)
- Reading and writing files
- Introduction to the **pandas** library

Install package from jupyter notebook



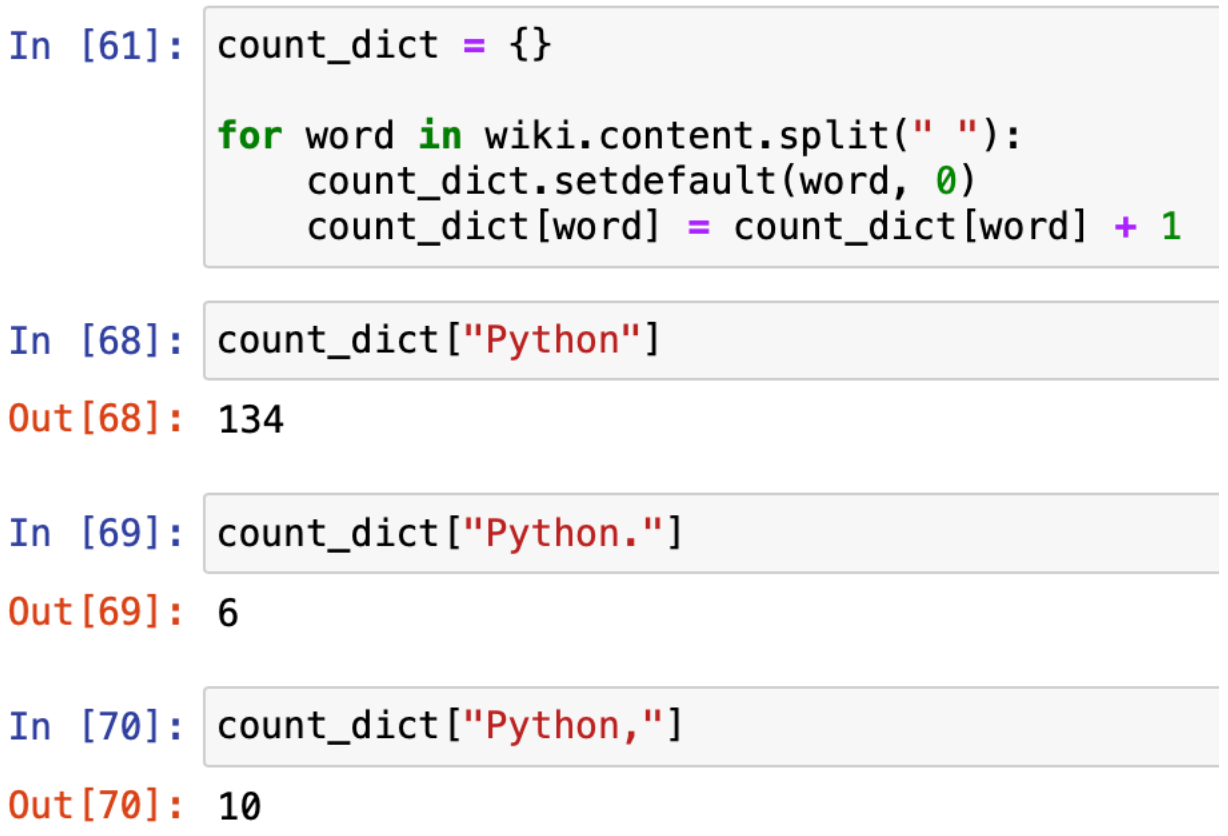
```
In [54]: !pip install wikipedia

Collecting wikipedia
  Downloading wikipedia-1.4.0.tar.gz (27 kB)
Requirement already satisfied: beautifulsoup4 in /Users/zeekresheff/opt/anaconda3/lib/python3.9/site-packages (from
wikipedia) (4.11.1)
Requirement already satisfied: requests<3.0.0,>=2.0.0 in /Users/zeekresheff/opt/anaconda3/lib/python3.9/site-packag
es (from wikipedia) (2.27.1)
Requirement already satisfied: idna<4,>=2.5 in /Users/zeekresheff/opt/anaconda3/lib/python3.9/site-packages (from r
equests<3.0.0,>=2.0.0->wikipedia) (3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in /Users/zeekresheff/opt/anaconda3/lib/python3.9/site-pac
kages (from requests<3.0.0,>=2.0.0->wikipedia) (2.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /Users/zeekresheff/opt/anaconda3/lib/python3.9/site-package
s (from requests<3.0.0,>=2.0.0->wikipedia) (1.26.9)
Requirement already satisfied: certifi>=2017.4.17 in /Users/zeekresheff/opt/anaconda3/lib/python3.9/site-packages (
from requests<3.0.0,>=2.0.0->wikipedia) (2021.10.8)
Requirement already satisfied: soupsieve>1.2 in /Users/zeekresheff/opt/anaconda3/lib/python3.9/site-packages (from
beautifulsoup4->wikipedia) (2.3.1)
Building wheels for collected packages: wikipedia
  Building wheel for wikipedia (setup.py) ... done
  Created wheel for wikipedia: filename=wikipedia-1.4.0-py3-none-any.whl size=11695 sha256=6cb4177dae0a6687b76396f2
a6d9f35300bb3d9f01ae9adf5ef37e7b9bb45e39
  Stored in directory: /Users/zeekresheff/Library/Caches/pip/wheels/c2/46/f4/caa1bee71096d7b0cdca2f2a2af45cacf35c57
60bee8f00948
Successfully built wikipedia
Installing collected packages: wikipedia
Successfully installed wikipedia-1.4.0
```

Get text from Wikipedia



```
In [55]: import wikipedia

In [57]: wiki = wikipedia.page('Python (programming language)')

In [58]: wiki.content

Out[58]: 'Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability wi
         th the use of significant indentation.Python is dynamically-typed and garbage-collected. It supports multiple progr
         amming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is
         often described as a "batteries included" language due to its comprehensive standard library.Guido van Rossum began
         working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as
         Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detect
         ing garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision
         that is not completely backward-compatible with earlier versions. Python 2 was discontinued with version 2.7.18 in
         2020.Python consistently ranks as one of the most popular programming languages.\n\n\n== History ==\n\nPython was c
         onceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a succ
         essor to the ABC programming language, which was inspired by SETL, capable of exception handling (from the start pl
         us new capabilities in Python 3.11) and interfacing with the Amoeba operating system. Its implementation began in D
         ecember 1989. Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018,
         when he announced his "permanent vacation" from his responsibilities as Python\'s "benevolent dictator for life", a
         title the Python community bestowed upon him to reflect his long-term commitment as the project\'s chief decision-m
         aker. In January 2019, active Python core developers elected a five-member Steering Council to lead the project.Pyt
         hon 2.0 was released on 16 October 2000, with many major new features. Python 3.0, released on 3 December 2008, wit
         h many of its major features backported to Python 2.6.x and 2.7.x.  Releases of Python 3 include the 2to3 utility,
         which automates the translation of Python 2 code to Python 3.Python 2.7\'s end-of-life was initially set for 2015,
```

Let’s count words!



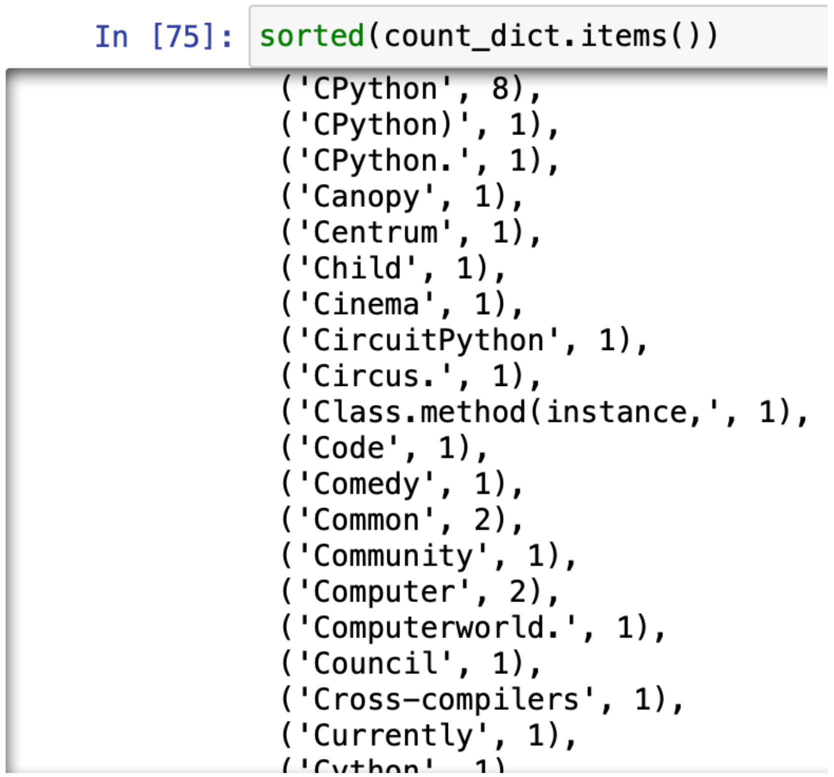
```
In [61]: count_dict = {}

         for word in wiki.content.split(" "):
             count_dict.setdefault(word, 0)
             count_dict[word] = count_dict[word] + 1

In [68]: count_dict["Python"]

Out[68]: 134

In [69]: count_dict["Python."]

Out[69]: 6

In [70]: count_dict["Python,"]

Out[70]: 10
```

Class exercise: count without the punctuation

- Use string functinos (such as .replace) to count the number of words regardless of:
  - Punctuation before/after the word
  - Upper or lower case

Sorting by key

- To sort a dictionary by key (words) we use:



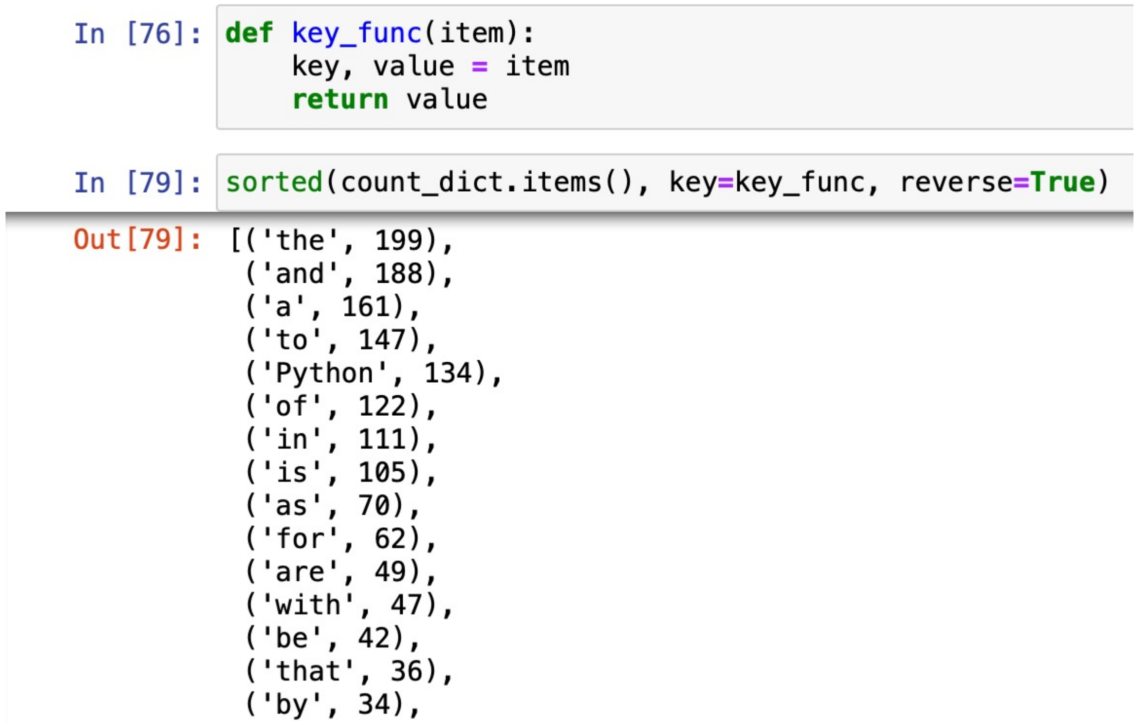
```
In [75]: sorted(count_dict.items())
```

```
('CPython', 8),
('CPython)', 1),
('CPython.', 1),
('Canopy', 1),
('Centrum', 1),
('Child', 1),
('Cinema', 1),
('CircuitPython', 1),
('Circus.', 1),
('Class.method(instance,', 1),
('Code', 1),
('Comedy', 1),
('Common', 2),
('Community', 1),
('Computer', 2),
('Computerworld.', 1),
('Council', 1),
('Cross-compilers', 1),
('Currently', 1),
('Cython', 1)
```

Sorting by value

- To sort a dictionary by value (word count) we need a more complicated functionality: a key function

- a mapping: object -> number

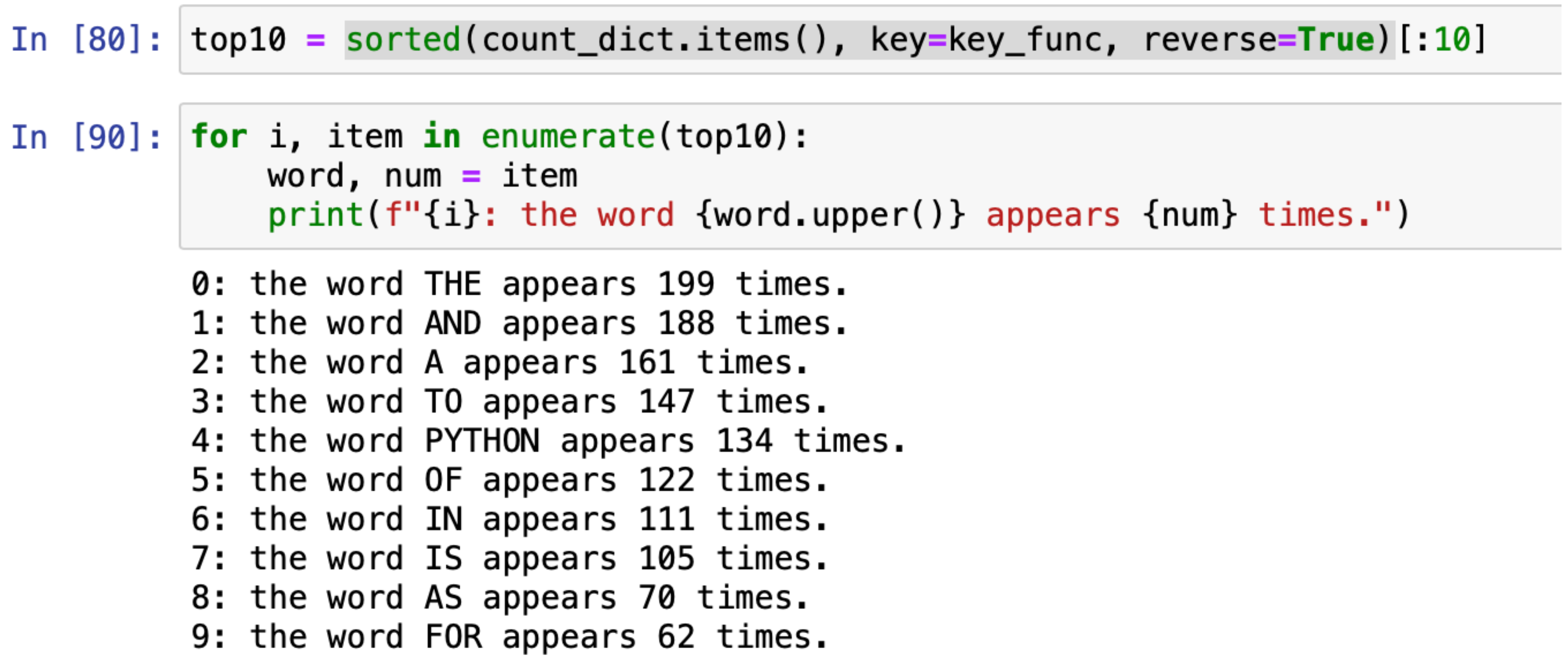


```
In [76]: def key_func(item):
             key, value = item
             return value

In [79]: sorted(count_dict.items(), key=key_func, reverse=True)

Out[79]: [('the', 199),
          ('and', 188),
          ('a', 161),
          ('to', 147),
          ('Python', 134),
          ('of', 122),
          ('in', 111),
          ('is', 105),
          ('as', 70),
          ('for', 62),
          ('are', 49),
          ('with', 47),
          ('be', 42),
          ('that', 36),
          ('by', 34),
```

Looping through Containers: Enumerate

The built-in function enumerate(seq) returns the position index and corresponding value at the same time.

Example:



```
In [80]: top10 = sorted(count_dict.items(), key=key_func, reverse=True)[:10]
```

```
In [90]: for i, item in enumerate(top10):
             word, num = item
             print(f"{i}: the word {word.upper()} appears {num} times.")
```

```
0: the word THE appears 199 times.
1: the word AND appears 188 times.
2: the word A appears 161 times.
3: the word TO appears 147 times.
4: the word PYTHON appears 134 times.
5: the word OF appears 122 times.
6: the word IN appears 111 times.
7: the word IS appears 105 times.
8: the word AS appears 70 times.
9: the word FOR appears 62 times.
```

Exercise

- Enumrate all the words that appear at least 25 times

A solution

```
In [91]: for i, item in enumerate(sorted(count_dict.items(), key=key_func, reverse=True)):
             word, num = item
             if num < 25:
                 break
             print(f"{i}: the word {word.upper()} appears {num} times.")
```

```
0: the word THE appears 199 times.
1: the word AND appears 188 times.
2: the word A appears 161 times.
3: the word TO appears 147 times.
4: the word PYTHON appears 134 times.
5: the word OF appears 122 times.
6: the word IN appears 111 times.
7: the word IS appears 105 times.
8: the word AS appears 70 times.
9: the word FOR appears 62 times.
10: the word ARE appears 49 times.
11: the word WITH appears 47 times.
12: the word BE appears 42 times.
13: the word THAT appears 36 times.
14: the word BY appears 34 times.
15: the word USED appears 31 times.
16: the word WHICH appears 30 times.
17: the word OR appears 29 times.
18: the word HAS appears 29 times.
19: the word FROM appears 28 times.
20: the word PROGRAMMING appears 26 times.
21: the word WAS appears 26 times.
22: the word ALSO appears 25 times.
```

Looping through Containers: Zip

Make an iterator that aggregates elements from each of the iterables. Returns an iterator of tuples, where the i-th tuple contains the i-th element from each of the argument sequences or iterables. The iterator stops when the shortest input iterable is exhausted.

```
In [147]: x = [1,2,3]
In [148]: y = [4,5,6]
In [149]: zip(x,y)
Out[149]: <zip at 0x111811f88>

In [150]: list(_)
Out[150]: [(1, 4), (2, 5), (3, 6)]

In [151]: questions = ['name', 'quest', 'favorite color']
In [152]: answers = ['lancelot', 'the holy grail', 'blue']

In [153]: for q, a in zip(questions, answers):
     ...:     print('What is your {0}?  It is {1}.'.format(q, a))
     ...:
What is your name?  It is lancelot.
What is your quest?  It is the holy grail.
What is your favorite color?  It is blue.
```

Unpacking tuples

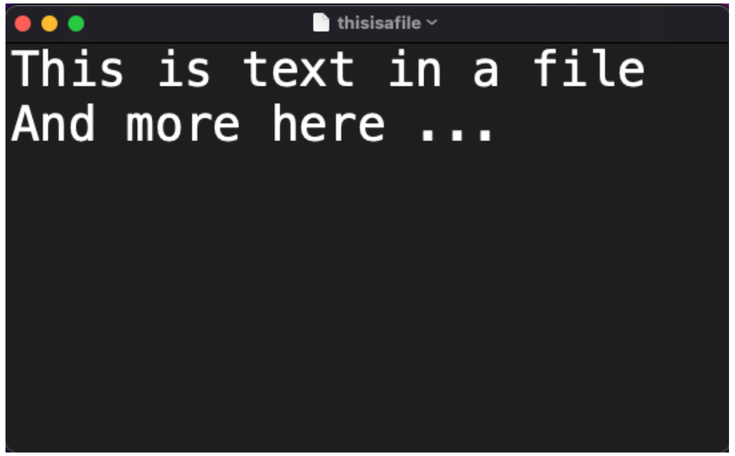
Comparing collections

type	What is it	Create	Orderd	Iterable	Mutable
tuple	Easy way to group multiple “things” in a single variable -- t = 1, 2, 3 -- t = (1,2,3) -- return a, b	a,b	Yes	Yes	No
list	Ordered collection of elements -- li = [1,2,3]	[a,b]	Yes	Yes	Yes
set	Unordered collection of elements without repeatition -- s = {1,2,3}	{a,b}	No	Yes	Yes
dict	Mapping from key to value -- d = {“a”: 1, “b”: 2}	{”key a”: “val a”}	No	Yes	Yes (values)

Reading from files

```
f = open("thisisafile", "r")

row1 = f.readline()
print(row1)

row2 = f.readline()
print(row2)

f.close()
```

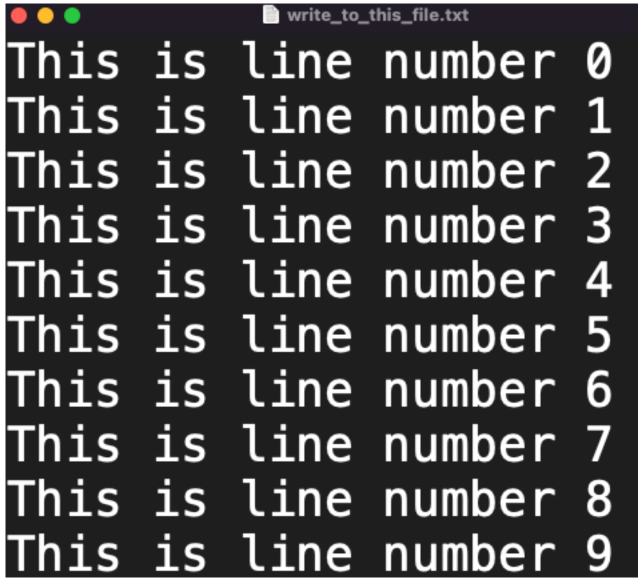
```
This is text in a file

And more here ...
```



```
This is text in a file
And more here ...
```

Writing to files

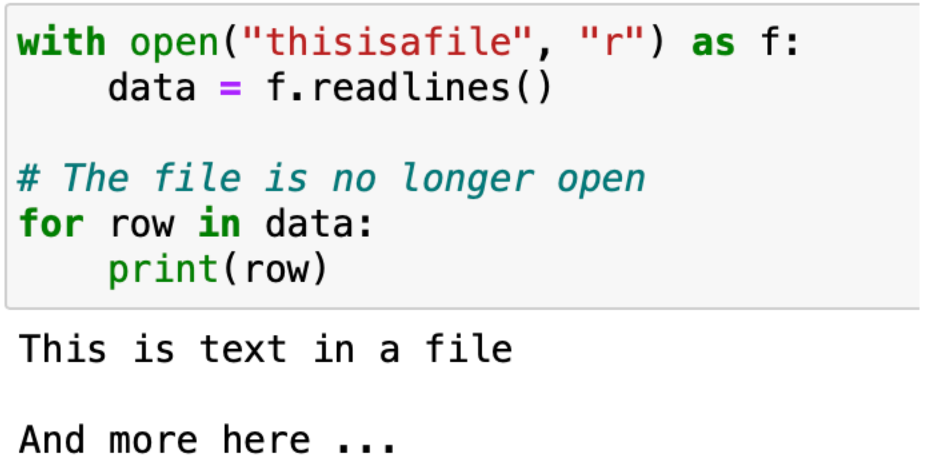
```python
write_file = open("write_to_this_file.txt", "w")

for i in range(10):
    s = f"This is line number {i}\n"
    write_file.write(s)

write_file.close()
```


write_to_this_file.txt

```
This is line number 0
This is line number 1
This is line number 2
This is line number 3
This is line number 4
This is line number 5
This is line number 6
This is line number 7
This is line number 8
This is line number 9
```

With block

- Replace the open-close logic:



```
with open("thisisafile", "r") as f:
    data = f.readlines()

# The file is no longer open
for row in data:
    print(row)
```

```
This is text in a file

And more here ...
```

Exercise

- Write a function:

```
def sum_from_file(file_name):
    …
```

that takes the name of a file, reads the file row by row, and adds up all rows that contain a number. For eample if the file contains:

```
bla bla this is not a number
8
3
bla bla  this is not a number either
```

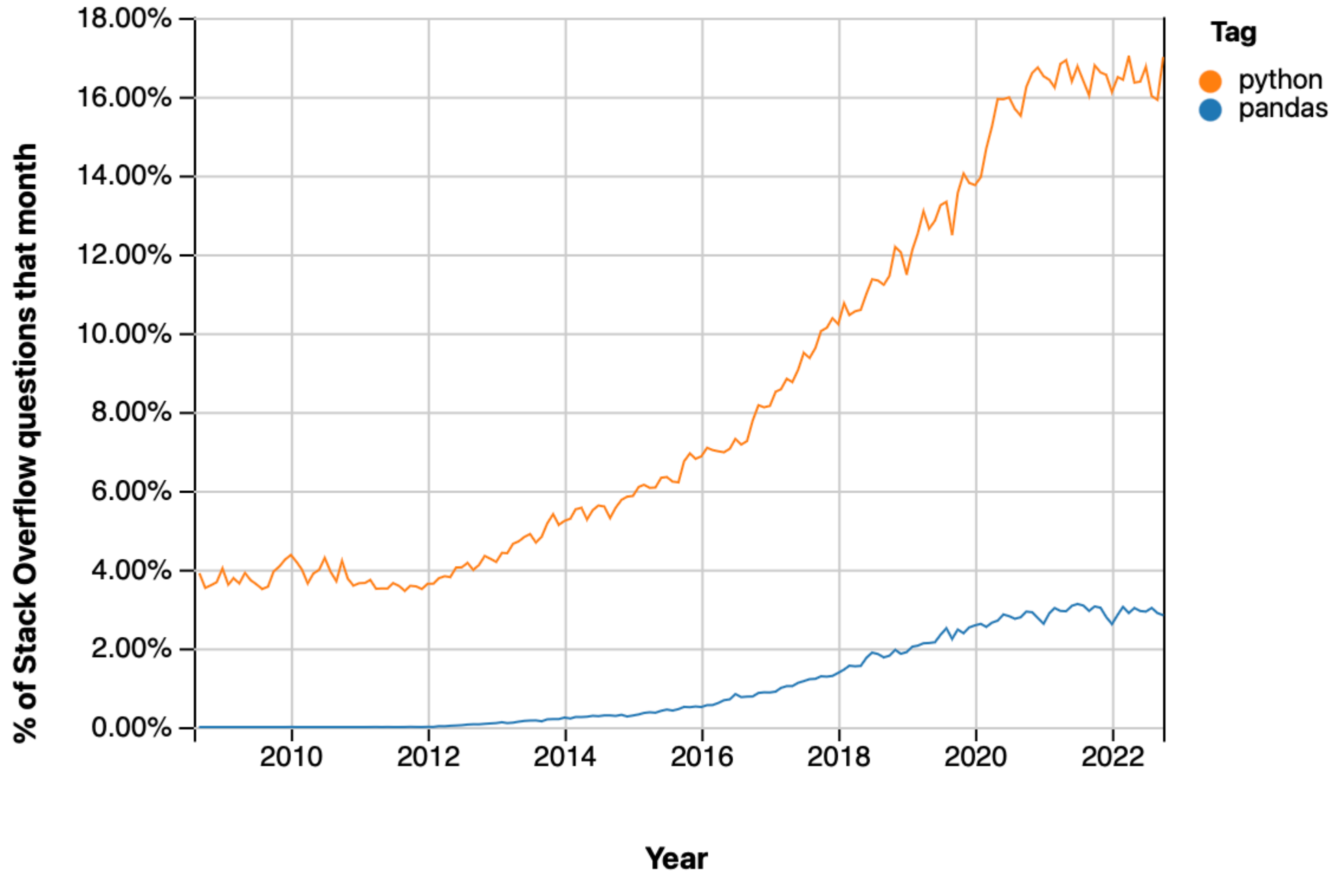
Then the output is 11.

Pandas

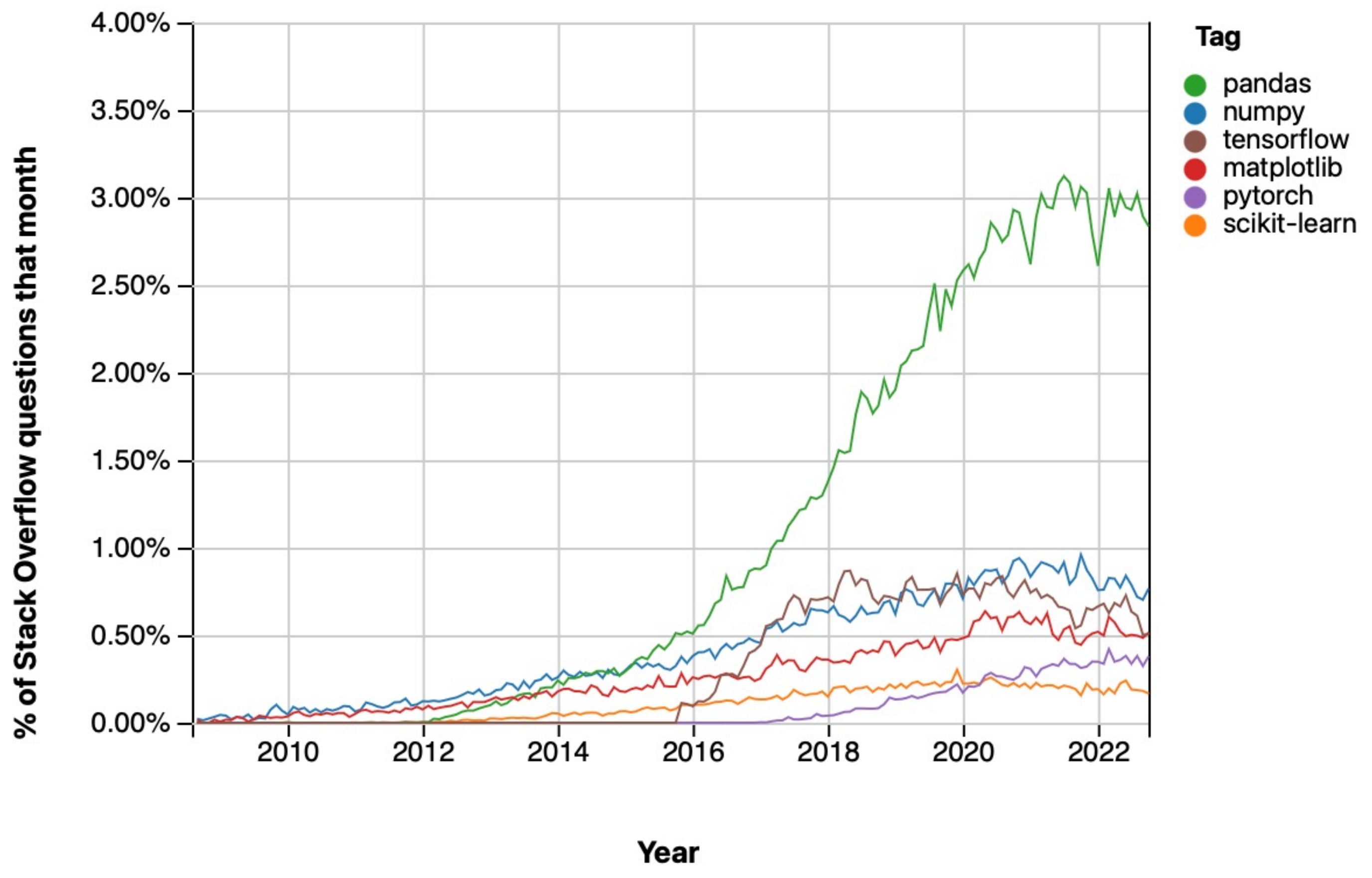
Pandas

- Panel data (observations over time/ Python data analysis)
- But now much more…
- Very popular data science tool
- Tabular data

Pandas popularity on the rise



Pandas vs. other data-science packages

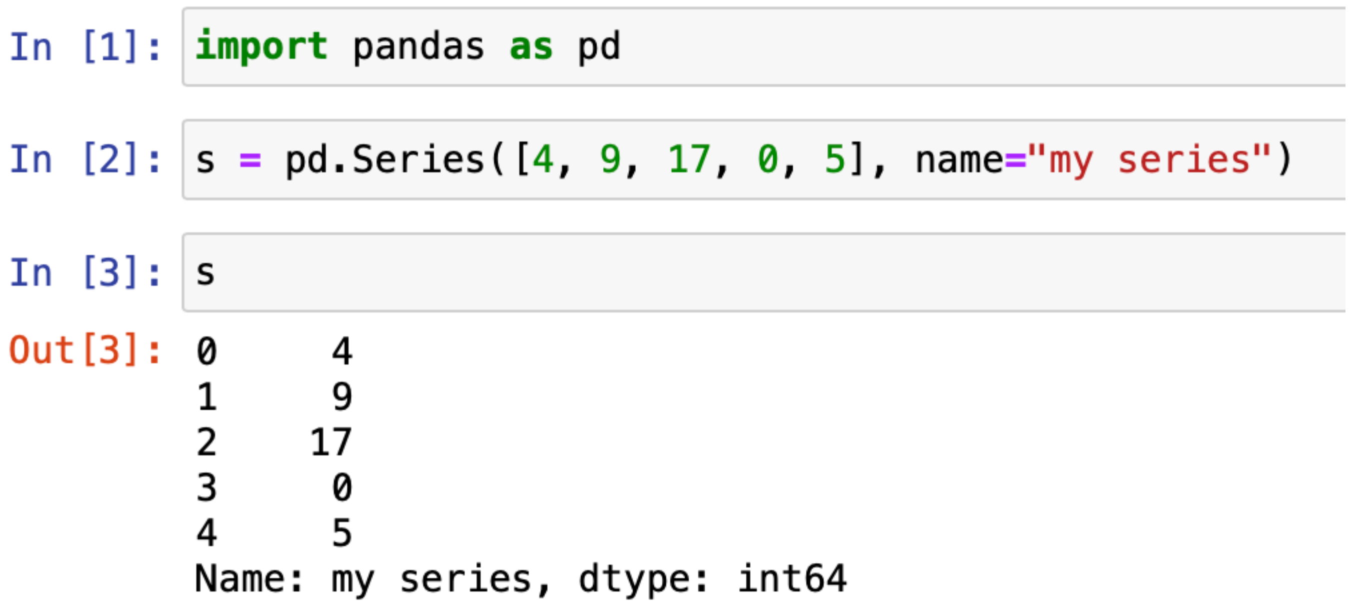


Pandas objects

- Series
  - Like a single excel column

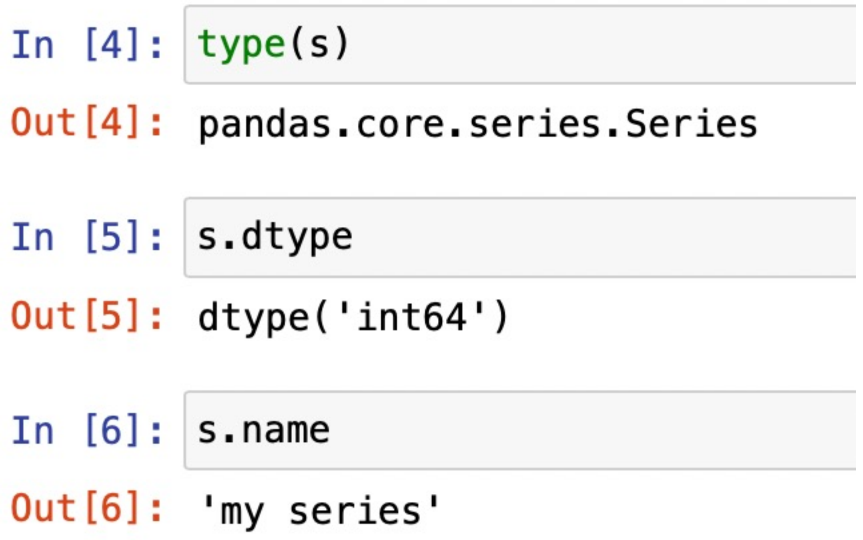
- DataFrame
  - Like a single spredsheet (each column is a Series)

Pandas Series



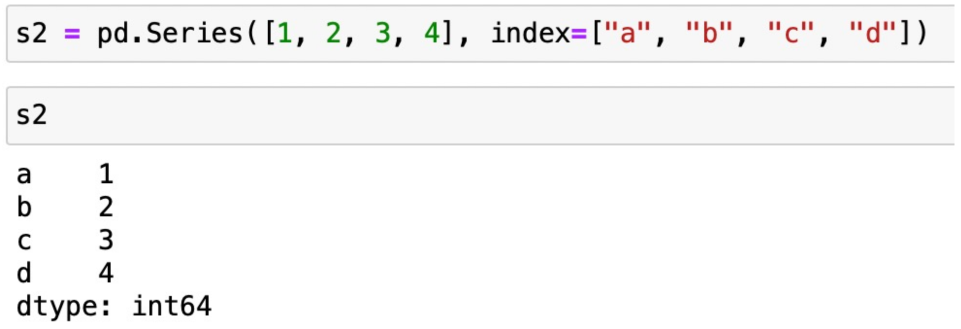
```
In [1]: import pandas as pd

In [2]: s = pd.Series([4, 9, 17, 0, 5], name="my series")

In [3]: s

Out[3]: 0     4
        1     9
        2    17
        3     0
        4     5
        Name: my series, dtype: int64
```

type, dtype, name



```
In [4]: type(s)
Out[4]: pandas.core.series.Series

In [5]: s.dtype
Out[5]: dtype('int64')

In [6]: s.name
Out[6]: 'my series'
```

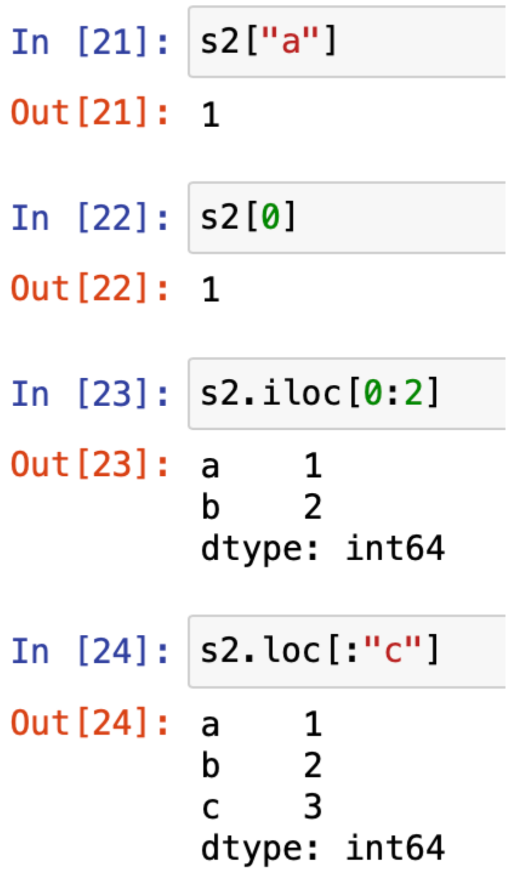
Series indexing



```
s2 = pd.Series([1, 2, 3, 4], index=["a", "b", "c", "d"])
```

```
s2
```

```
a    1
b    2
c    3
d    4
dtype: int64
```

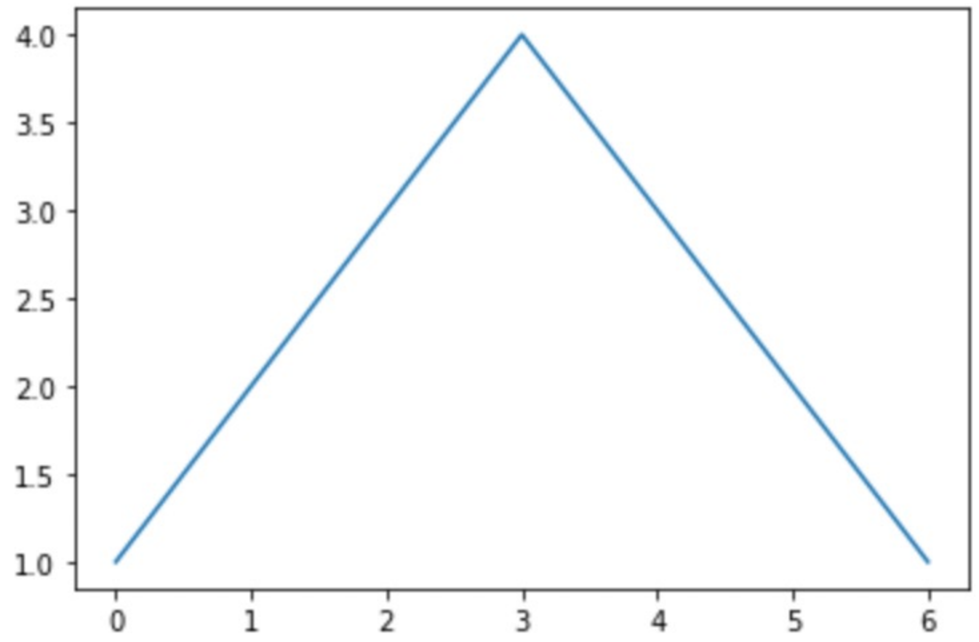


```
In [21]: s2["a"]

Out[21]: 1

In [22]: s2[0]

Out[22]: 1

In [23]: s2.iloc[0:2]

Out[23]: a    1
         b    2
         dtype: int64

In [24]: s2.loc[:"c"]

Out[24]: a    1
         b    2
         c    3
         dtype: int64
```

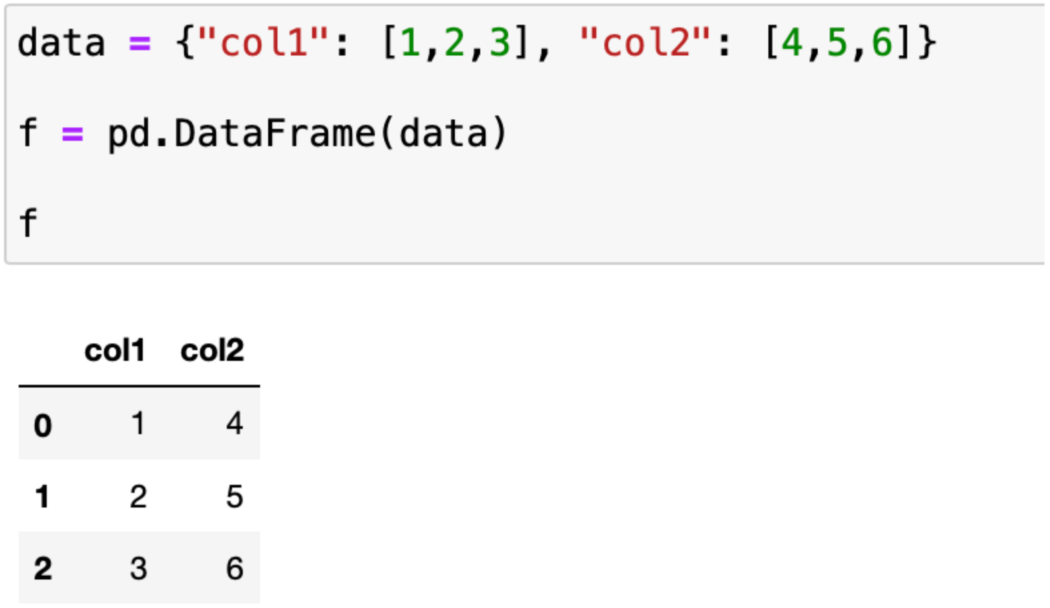
Plot

```
s3 = pd.Series([1, 2, 3, 4, 3, 2, 1])
s3.plot()
```

```
<AxesSubplot:>
```

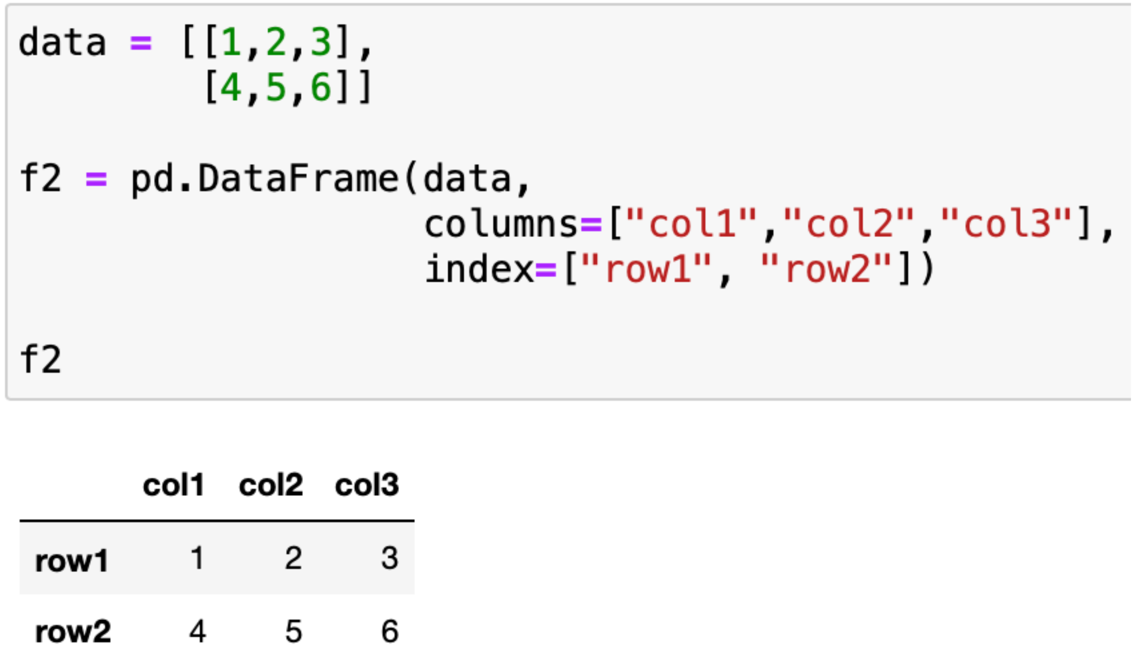


Pandas DataFrame



```python
data = {"col1": [1,2,3], "col2": [4,5,6]}

f = pd.DataFrame(data)

f
```

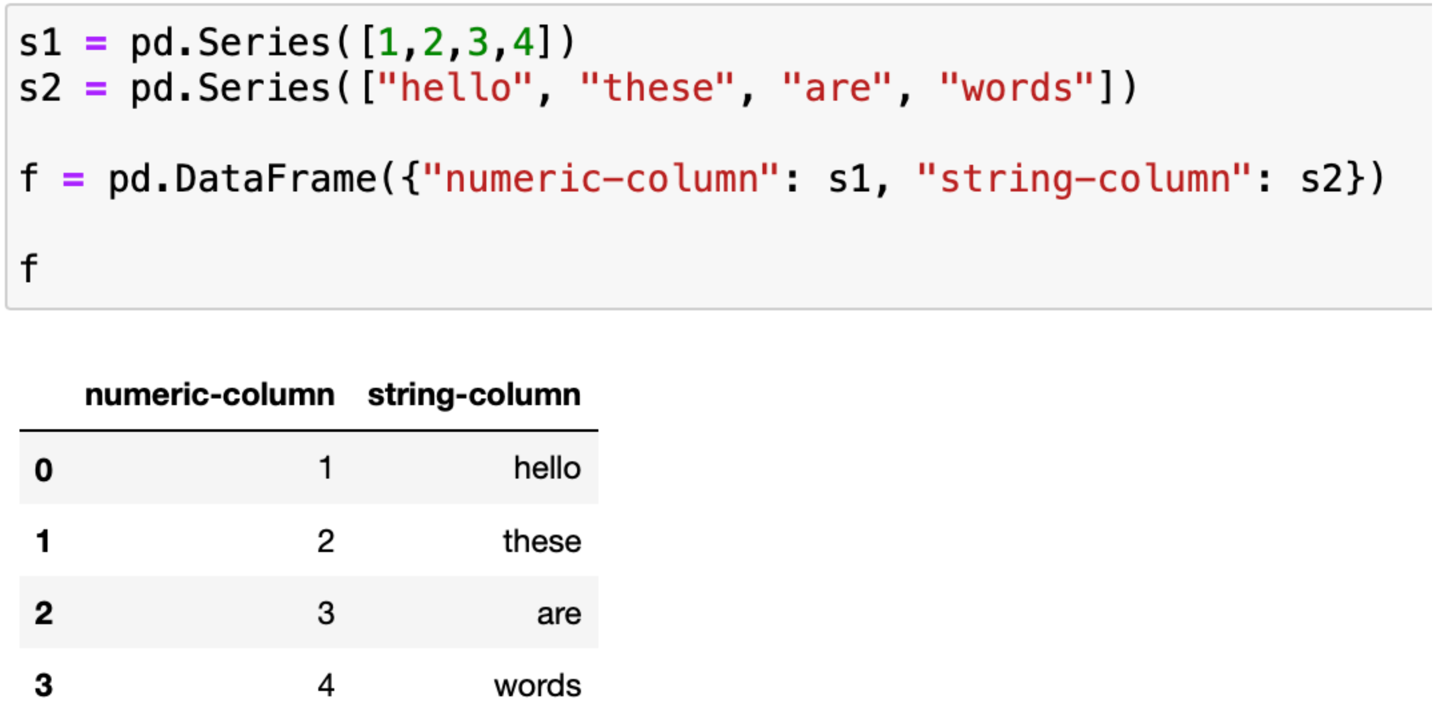
	col1	col2
0	1	4
1	2	5
2	3	6



```python
data = [[1,2,3],
        [4,5,6]]

f2 = pd.DataFrame(data,
                  columns=["col1","col2","col3"],
                  index=["row1", "row2"])

f2
```

	col1	col2	col3
row1	1	2	3
row2	4	5	6

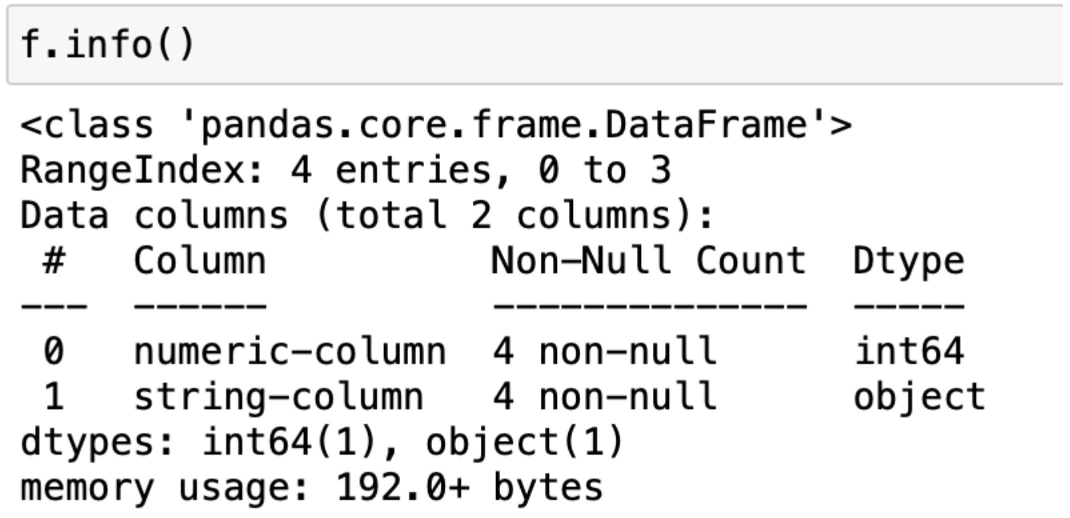
DataFrame from multiple Series



```
s1 = pd.Series([1,2,3,4])
s2 = pd.Series(["hello", "these", "are", "words"])

f = pd.DataFrame({"numeric-column": s1, "string-column": s2})

f
```

	numeric-column	string-column
0	1	hello
1	2	these
2	3	are
3	4	words

info



```
f.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4 entries, 0 to 3
Data columns (total 2 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   numeric-column  4 non-null      int64
 1   string-column   4 non-null      object
dtypes: int64(1), object(1)
memory usage: 192.0+ bytes
```

DataFrame column and index

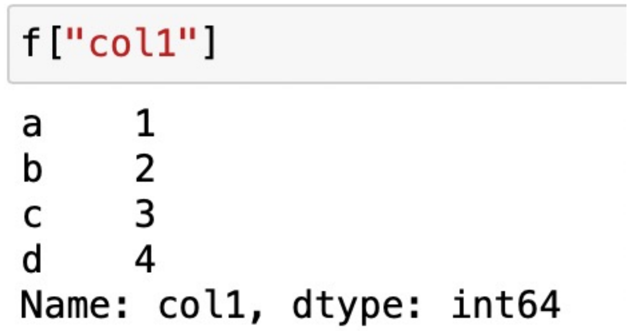
```
f.index = ["a", "b", "c", "d"]
f
```

	numeric-column	string-column
a	1	hello
b	2	these
c	3	are
d	4	words

```
f.columns = ["col1", "col2"]
f
```

	col1	col2
a	1	hello
b	2	these
c	3	are
d	4	words

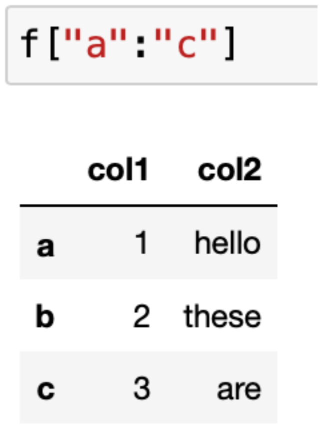
DataFrame indexing



```
f["col1"]
```

```
a    1
b    2
c    3
d    4
Name: col1, dtype: int64
```

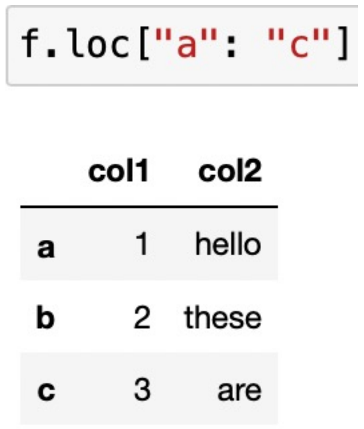
get column (Series)



```
f["a":"c"]
```

	col1	col2
a	1	hello
b	2	these
c	3	are

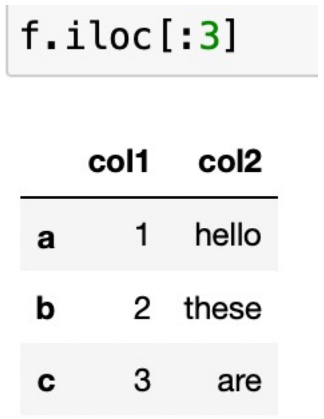
rows by index



```
f.loc["a": "c"]
```

	col1	col2
a	1	hello
b	2	these
c	3	are

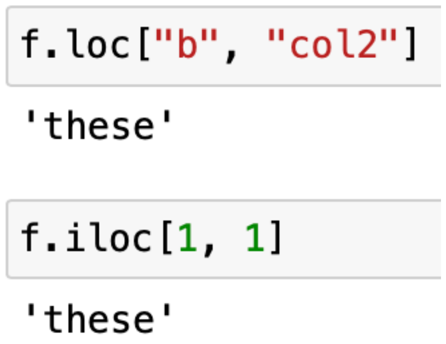
rows by index



```
f.iloc[:3]
```

	col1	col2
a	1	hello
b	2	these
c	3	are

rows by position



```
f.loc["b", "col2"]
```

```
'these'
```

```
f.iloc[1, 1]
```

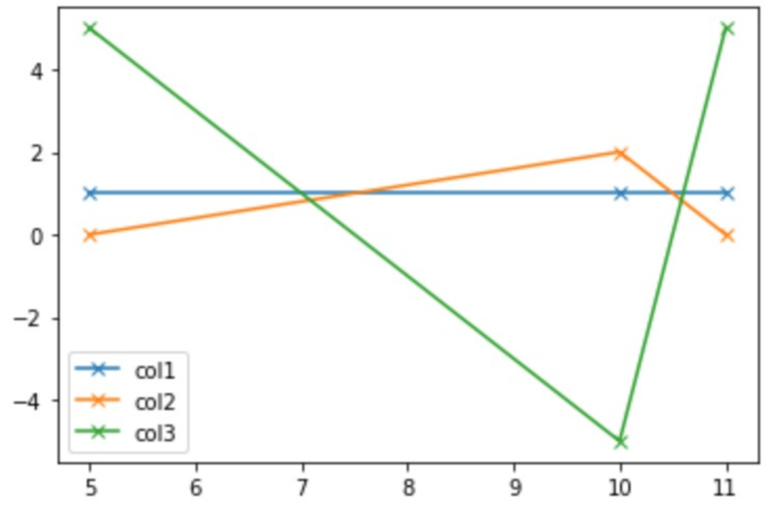
```
'these'
```

cell by index/column

Plot

```python
data = [[1, 0, 5],
        [1, 2, -5],
        [1, 0, 5]]

f = pd.DataFrame(data,
                 columns=["col1","col2","col3"]
                 index=[5, 10, 11])

f.plot(style="-x")
```

```
<AxesSubplot:>
```



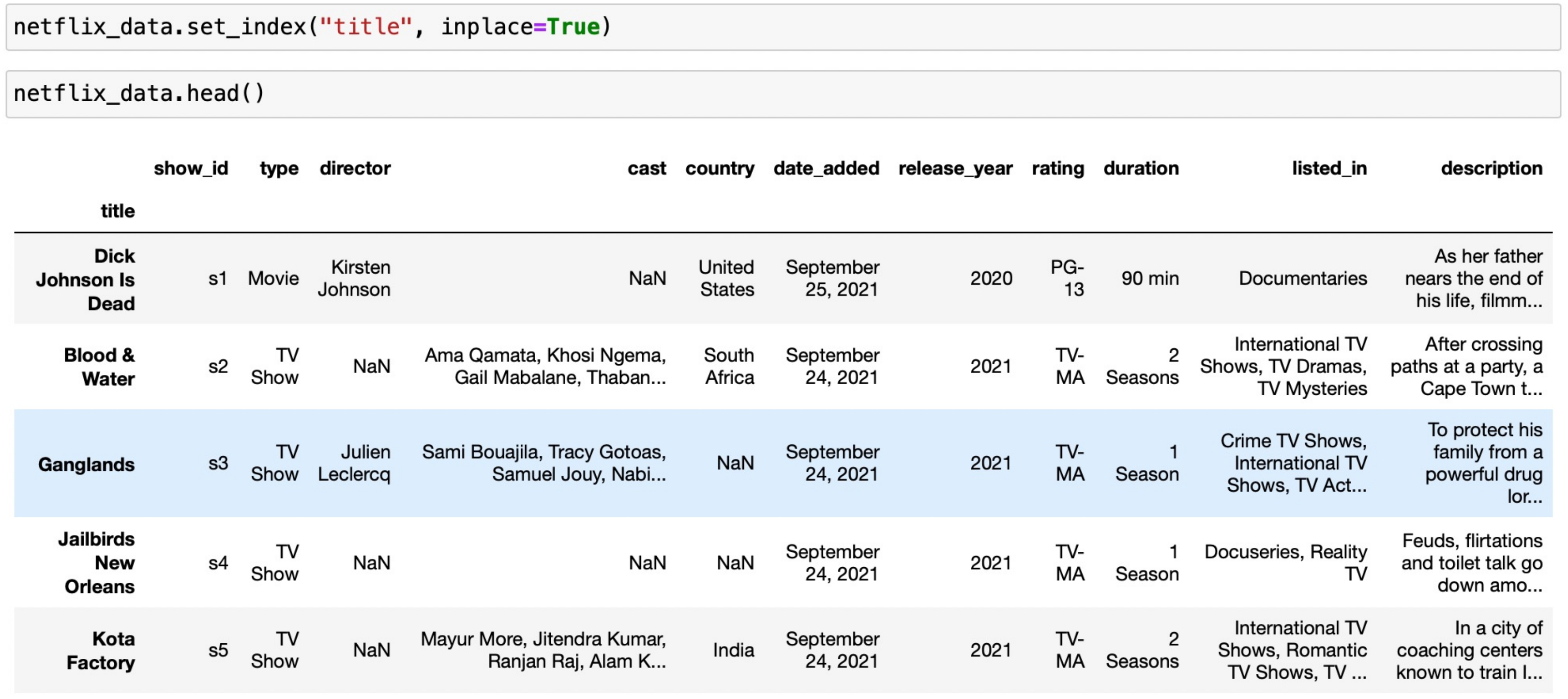
read_csv

```python
netflix_data = pd.read_csv("~/Desktop/netflix_titles.csv")
```

```python
netflix_data.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...

Make the title column the index



```
netflix_data.set_index("title", inplace=True)
```

```
netflix_data.head()
```

title	show_id	type	director	cast	country	date_added	release_year	rating	duration	listed_in	description
Dick Johnson Is Dead	s1	Movie	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
Blood & Water	s2	TV Show	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
Ganglands	s3	TV Show	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
Jailbirds New Orleans	s4	TV Show	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
Kota Factory	s5	TV Show	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train l...

Class exercise

- load the data using rad_csv
- make the movie title the index using set_index
- print the description field of ”The Matrix”

Filtering a DataFrame

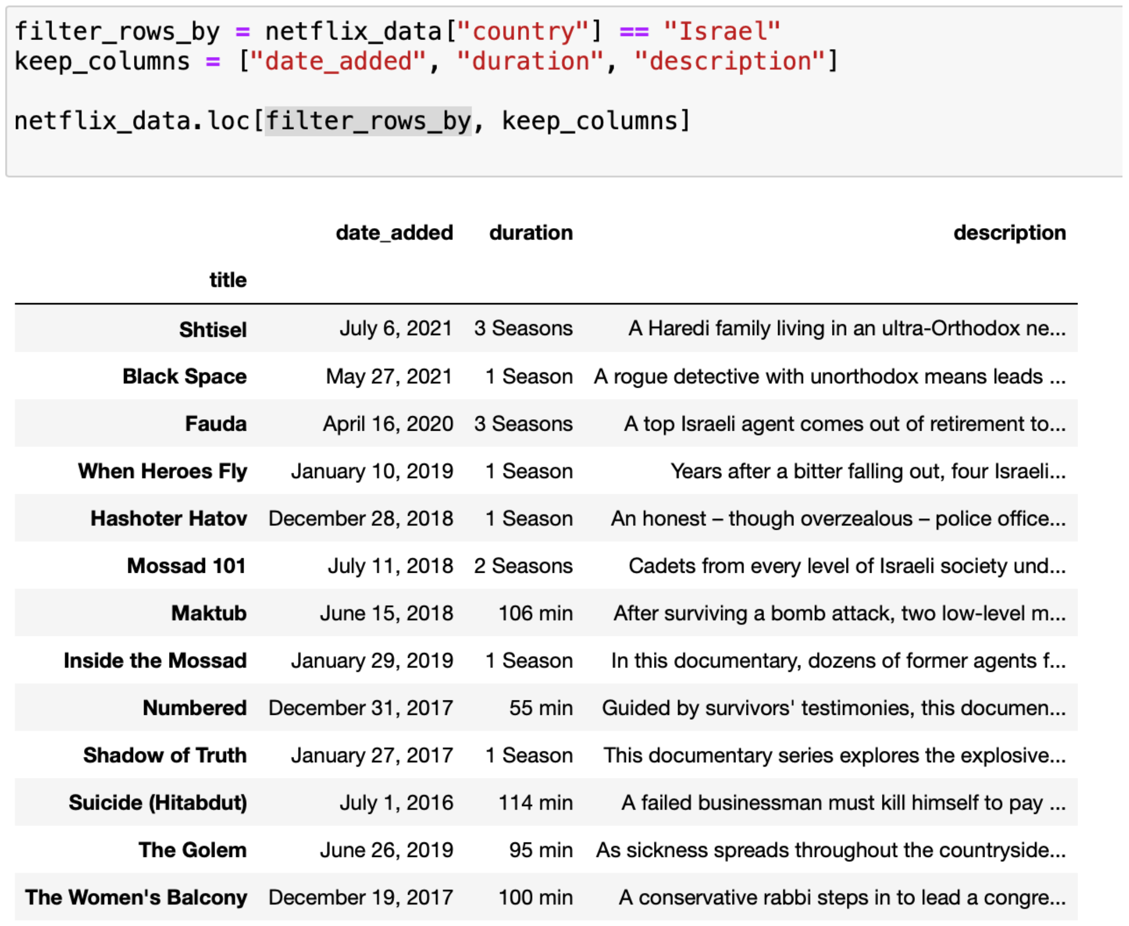
- What if I only want rows of Movies (no TV)?

```
filter_by = netflix_data["type"] == "Movie"

netflix_movies_only = netflix_data.loc[filter_by]
```

```
netflix_movies_only.sample(3)
```

	show_id	type	director	cast	country	date_added	release_year	rating	duration	listed_in	description
title											
God Calling	s1108	Movie	BB Sasore	Zainab Balogun, Karibi Fubara, Diana Egwuatu, ...	Nigeria	April 2, 2021	2018	TV-MA	120 min	Dramas, Faith & Spirituality, International Mo...	The loss of a child plunges Sade into a suicid...
My Happy Family	s5144	Movie	Nana Ekvtimishvili, Simon Gross	Ia Shugliashvili, Merab Ninidze, Berta Khapava...	Georgia, Germany, France	December 1, 2017	2017	TV-14	119 min	Dramas, Independent Movies, International Movies	A middle-aged wife and mother of two shocks he...
Katti Batti	s4449	Movie	Nikhil Advani	Imran Khan, Kangana Ranaut, Abhishek Saha, Mit...	India	November 1, 2018	2015	TV-14	132 min	Comedies, Dramas, International Movies	After falling in love during college, architec...

Filter rows and select columns



```
filter_rows_by = netflix_data["country"] == "Israel"
keep_columns = ["date_added", "duration", "description"]

netflix_data.loc[filter_rows_by, keep_columns]
```

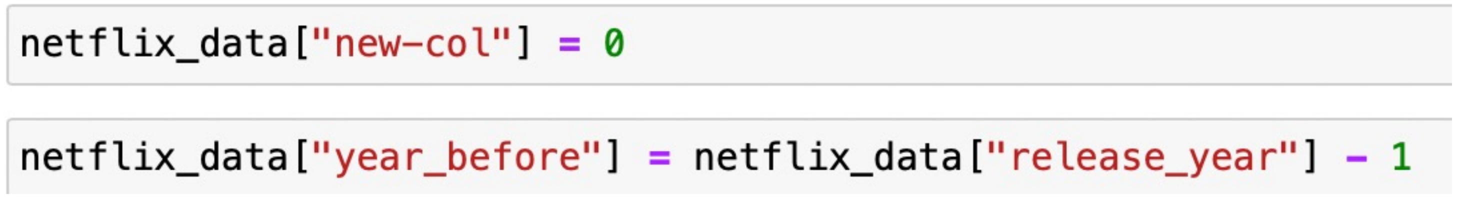
	date_added	duration	description
title			
Shtisel	July 6, 2021	3 Seasons	A Haredi family living in an ultra-Orthodox ne...
Black Space	May 27, 2021	1 Season	A rogue detective with unorthodox means leads ...
Fauda	April 16, 2020	3 Seasons	A top Israeli agent comes out of retirement to...
When Heroes Fly	January 10, 2019	1 Season	Years after a bitter falling out, four Israeli...
Hashoter Hatov	December 28, 2018	1 Season	An honest – though overzealous – police office...
Mossad 101	July 11, 2018	2 Seasons	Cadets from every level of Israeli society und...
Maktub	June 15, 2018	106 min	After surviving a bomb attack, two low-level m...
Inside the Mossad	January 29, 2019	1 Season	In this documentary, dozens of former agents f...
Numbered	December 31, 2017	55 min	Guided by survivors' testimonies, this documen...
Shadow of Truth	January 27, 2017	1 Season	This documentary series explores the explosive...
Suicide (Hitabdut)	July 1, 2016	114 min	A failed businessman must kill himself to pay ...
The Golem	June 26, 2019	95 min	As sickness spreads throughout the countryside...
The Women's Balcony	December 19, 2017	100 min	A conservative rabbi steps in to lead a congre...

Class exercise

- Find all the TV shows from Finland
- Print 2 rows of Icelandic TV shows or movies
- In which year were more titles released, 2019 or 2020?

Adding columns to a DataFrame

- Why add columns? Like in Excel… as part of a computation, as output



```
netflix_data["new-col"] = 0
```

```
netflix_data["year_before"] = netflix_data["release_year"] - 1
```

save DataFrame as csv

- my_frame.to_csv(path)

Exercise

- Write code that goes over the netflix data dataframe, and prints only movies where the description contains the name of the movie.
- What if we wanted to count them instead of print?